An Operations System for A Planetary Rover Prototype

Paul G. Backes*, Kam S. Tso†, Greg Tharp† and Won S. Kim*
*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California
†IA Tech Inc., Los Angeles, California

Abstract

A distributed operations system, the Web Interface for Telescience (WITS), is being developed for use in future planetary lander and rover mission operations. WITS provides visualization of downlink data and command sequence generation. WITS is an Internet-based tool written in the Java language and supports collaborative use by geographically distributed users over the Internet. WITS was used in the April 1999 Field Integration, Development, and Operations (FIDO) rover field test at Silver Lake, California. This paper describes the motivation for WITS, features, implementation, results from the field test, and plans for future work.

1 Introduction

The Web Interface for Telescience (WITS) is being developed for use in future planetary lander and rover mission operations. WITS provides an integrated operations environment for operations at the central operations location, collaboration by geographically distributed scientists, and public outreach. Collaboration by geographically distributed scientists enables participation in planetary rover mission operations by a greater number of scientists and reduces operations costs. Providing the same operations tool, with real mission data, to the public enables the public to have a better understanding of the mission and a more engaging mission experience. An earlier version of WITS provided some of the features needed for a planetary rover mission [1]. Based upon experience with WITS as an evaluation and public outreach tool in the 1997 Mars Pathfinder mission [2] and past field tests [3], WITS has been redesigned and reimplemented to provide all the features needed for planetary mission operations. This paper describes the new WITS system. The architecture is described in Section 2. Downlink data visualization is described in Section 3. Sequence generation is described in in Section 4. Distributed operations is described in Section 5 and Section 6 describes the implementation. Lessons learned from the April 1999 FIDO rover field test are discussed in Section 7. Future development plans are described in Section 8 and conclusions are given in Section 9.

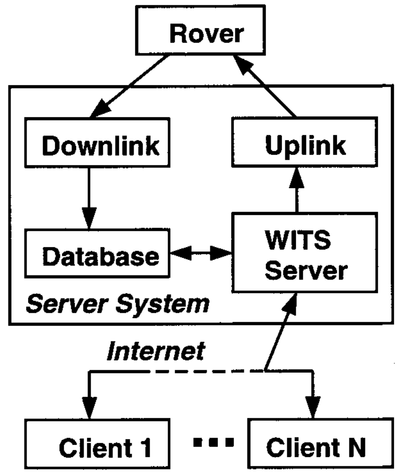

Figure 1: System Architecture

The Field Integration, Development, and Operations (FIDO) rover has been developed as a prototype vehicle for testing new planetary rover technologies and testing operations scenarios for the Mars'03 Athena rover mission [4]. The FIDO rover is similar in size and functionality to the Mars'03 mission Athena rover. A two week field test was performed with the FIDO rover at Silver Lake in the California Mojave desert in April 1999. The operational capabilities of WITS are described in this paper using examples from the April 1999 FIDO field test. The figures are of data from the field test.

2 System Architecture

The overall system architecture is shown in Figure 1. Variations on the architecture shown in Figure 1 are used depending on whether the system is being used to command a terrestrial prototype rover or a future planetary rover. In both cases, the Downlink system receives downlink data from the rover, processes the data, and puts the resulting data products into the Database. Multiple Client systems can be downloaded and run from anywhere on the Internet. The Client systems communicate with a Server system to read and write to the common Database. The Uplink system sends the final command sequence to the Rover. For the FIDO field test, a Client user instructed the Server to send the command sequence to the Uplink communication system which sent it to the rover. Early examples of Internet-based robot operation can be found in [5, 6].

In an actual planetary mission, various systems are involved in the uplink process, e.g., sequence finalization, checking and communication. Two architectural approaches are being considered for the Mars'03 rover mission. One approach would have WITS used by distributed scientist users to generate their sequence requests and submit them to JPL where the core operations team would integrate them into a final sequence using the Rover Control Workstation (RCW). The RCW would be an updated version of the system used in the 1997 Mars Pathfinder mission [7], but would not support distributed operations. Another approach under consideration is to integrate the high fidelity visualization features of the RCW with WITS to provide an integrated rover operations system. Then the WITS sequence generation capabilities described in this paper would be used by the core operations team for generating the final sequence as well. The RCW high fidelity visualization capabilities would run on dedicated high-end workstations.

3 Downlink Data Visualization

Downlink data from the rover is provided to the user via various views. The Results Tree window, as shown in Figure 2, displays the available downlink data for the mission and enables the user to select any downlink data to be visualized. The data is displayed in a tree structure whose nodes can be expanded or collapsed. Under results there are aerial and sites children. Aerial data is taken above the surface, e.g., by an orbiter or during descent to the surface. Sites are the places that a lander is at or a rover has traversed to. There can be various positions at each site. For a lander there is only one site and one position. The user opens a view to visualize downlink data by clicking on the item in the Results Tree. The various types of views are described below.

The Descent view provides images taken from an orbiter or from the spacecraft during descent to the surface. Clicking on a point in a Descent view causes the coordinates of the selected point to be drawn on the view. For a rover mission, the sites where the rover has been are displayed in the Descent views.

The Overhead view, shown in Figure 2, shows the area around the lander or rover from above. An optional grid provides angle and range information. Color-coded elevation map and texture mapped image options are provided. Targets and waypoints are displayed in the Overhead view, as well as selected points. Figure 2 shows a sparse color-coded elevation map. The black areas are areas where no range information is known.

The Panorama view, shown in Figure 3, is a mosaic of images taken by a stereo camera. Selecting a point in an image causes the x,y,z position on the surface to be displayed as well as the ARZ (azimuth angle, range from site center, and z elevation value). The point can be turned into a rover waypoint or a science target via the Main window Action pull-down menu. The Panorama view can be shown in 1/4, 1/2, and full scale.

The Wedge view, shown in Figure 2, displays one image with various options, e.g., left or right image from stereo image pair, various wavelengths, anaglyph, and resize. When a user selects a pixel in a Wedge or Panorama image, WITS determines the corresponding x,y,z position and surface normal on the terrain and displays the x,y,z position to the user in all views in which this point is visible. This point can be turned into a target or waypoint, as described below in Section 4. In the Wedge view, the pixel intensity is also displayed. In the Panorama view, the azimuth angle, range, and z value are also displayed.

The Contrast Adjuster view (selected from a Wedge view pull-down menu), shown in Figure 4, enables the contrast to be adjusted for a Wedge view image. The minimum and maximum desired pixel intensities are selected via scroll bars and then the pixel intensity values of the image are linearly stretched to have the selected pixel intensities become minimum (0) and maximum (255). The histogram of the initial and adjusted images are also shown.

The Instrument Data view, shown in Figure 5, provides a plot of instrument data. On the left is the columnized data and on the right is the plot of the data. The figure shows data from the point spectrometer on the FIDO rover.

4 Command Sequence Generation

The views discussed above provide a means for vi-

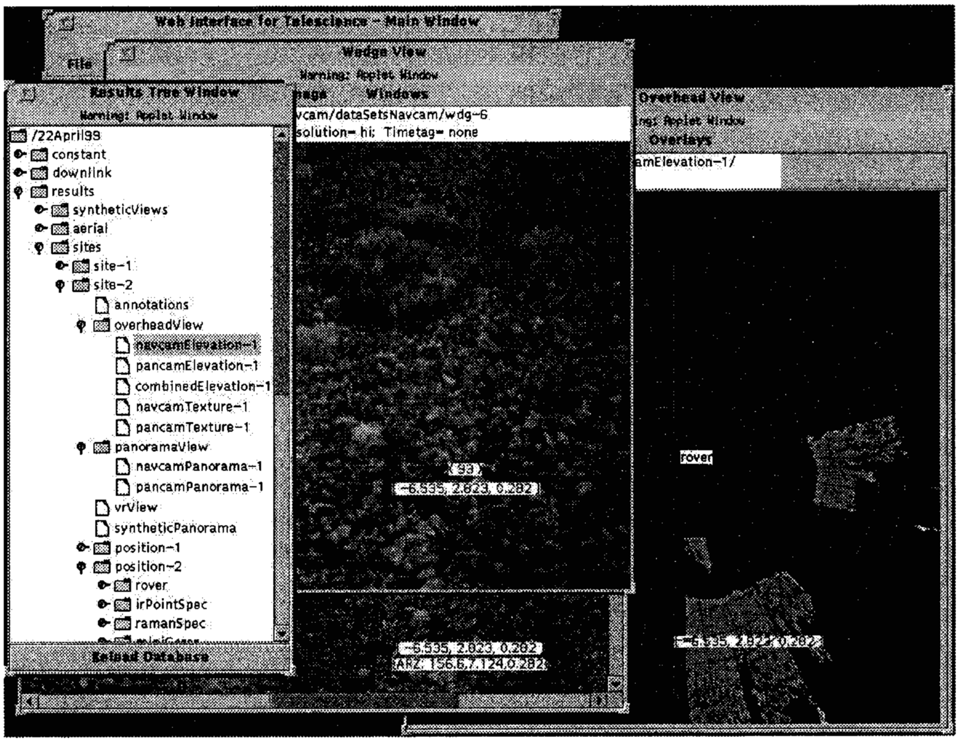

Figure 2: Results Tree, Wedge View, Overhead View

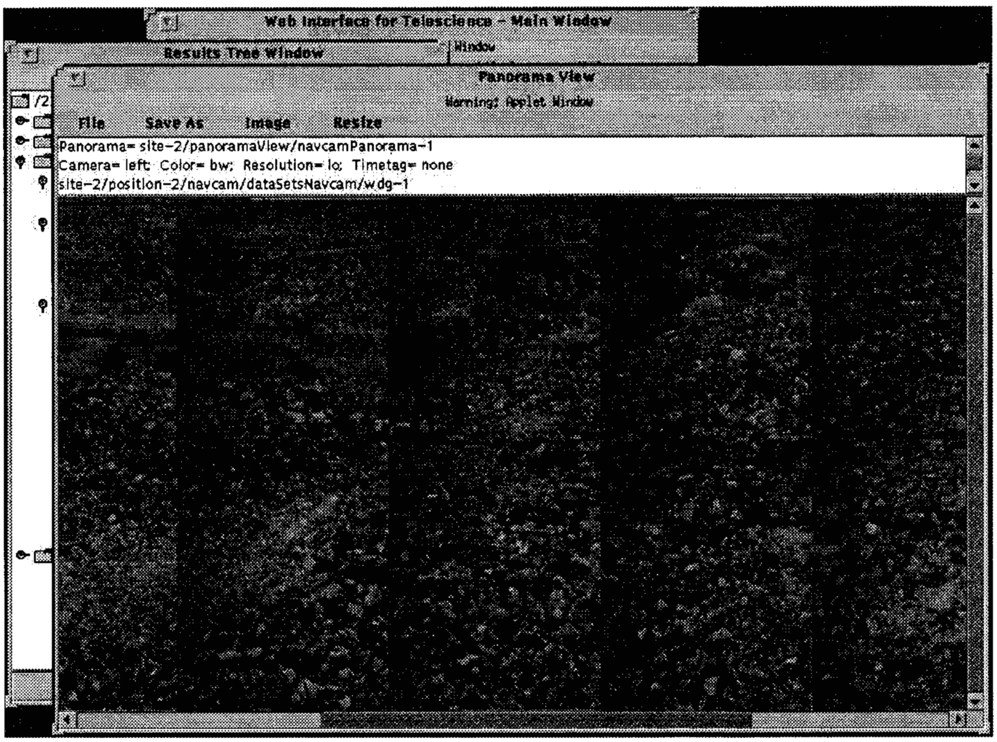

Figure 3: Panorama View

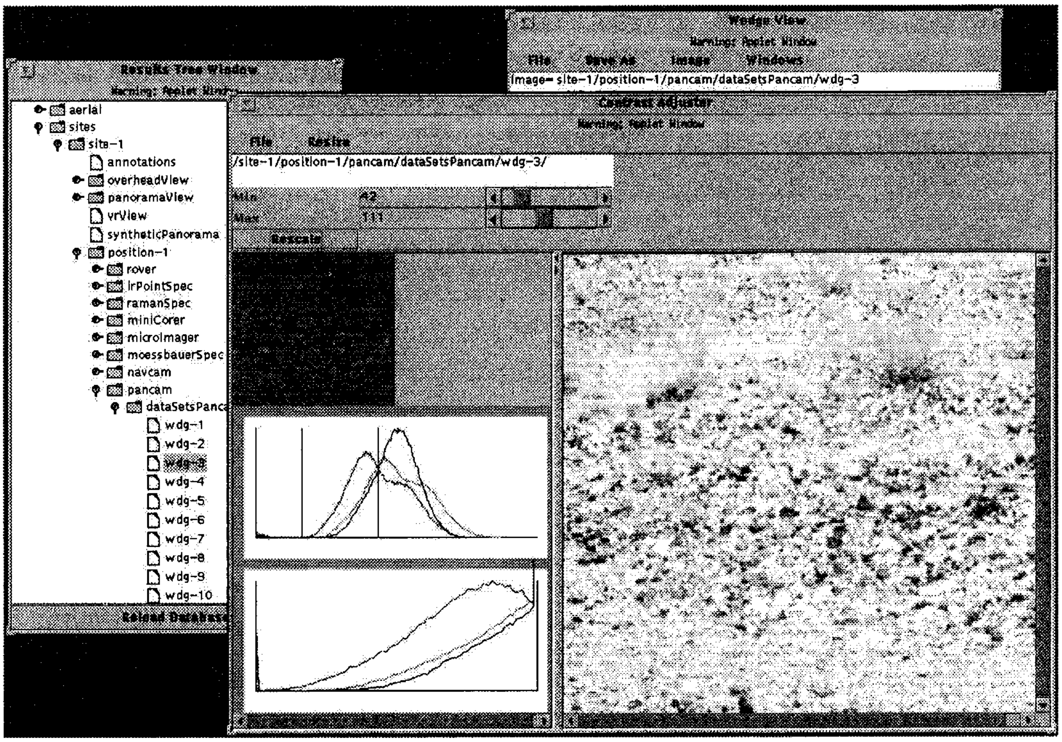

Figure 4: Contrast Adjuster View

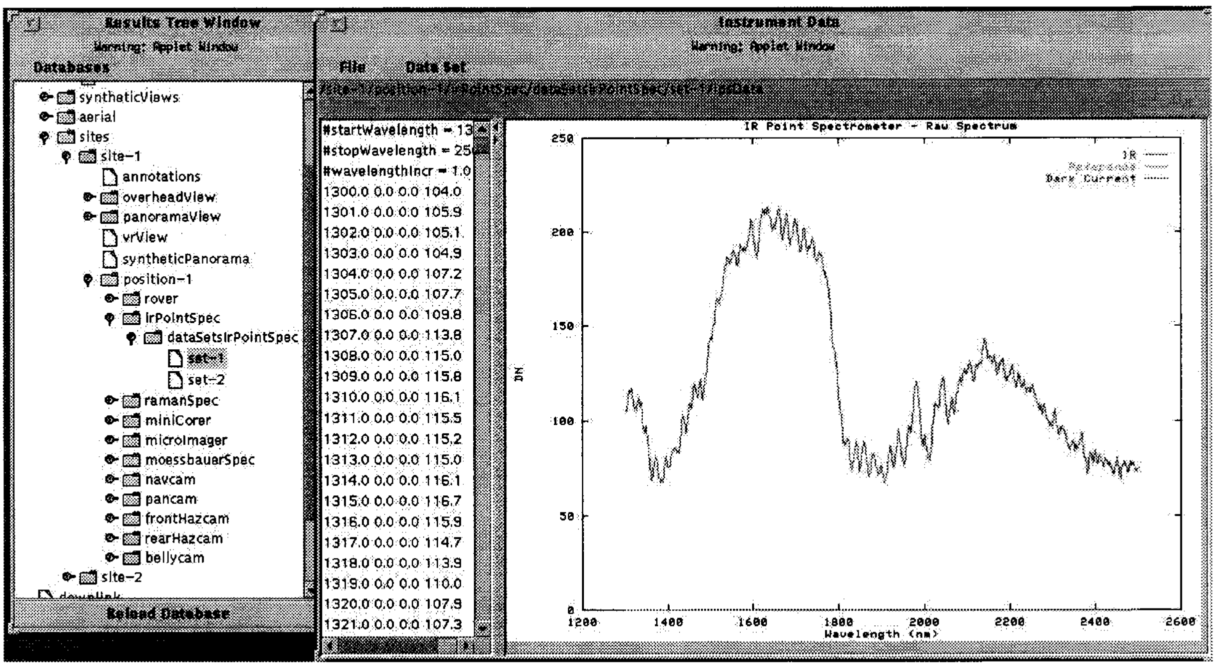

Figure 5: Instrument Data View

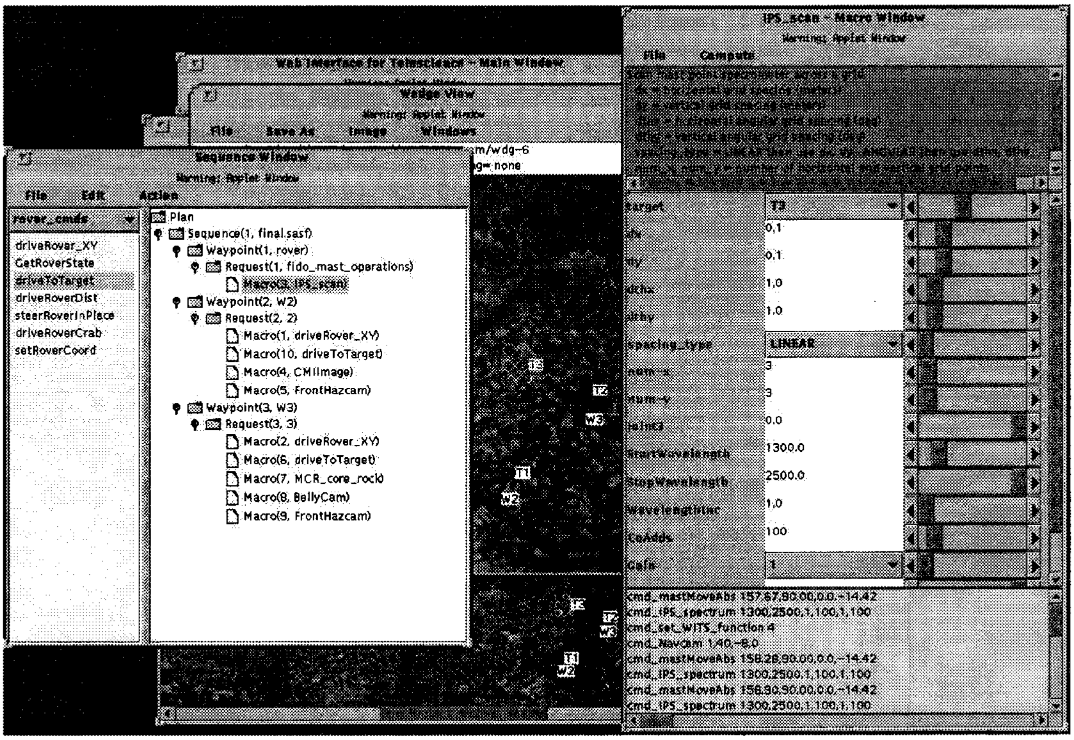

Figure 6: Sequence and Macro Windows

sualizing downlink mission data. WITS also provides various windows and features for command sequence generation, some of which are shown in Figure 6.

WITS enables 3D locations to be used as parameters in commands. As described in the Wedge view description above, a user specifies a 3D point by selecting a pixel in an image in either the Wedge or Panorama views and WITS determines the 3D coordinates and displays them at the selected point. The point can be turned into a target or waypoint by selecting "Add Target" or "Add Waypoint" in the Main window Action pull-down menu. Targets are pink circles and waypoints are blue squares. Waypoints are locations which the rover will drive to. Targets are general 3D points. WITS keeps track of all specified targets and waypoints and provides them as possible input parameters to sequence commands. In Figure 6, T2 is a target and W3 is a waypoint.

The Sequence window is used to generate a command sequence. A command sequence has a hierarchy of elements. The hierarchy, in descending order, is: Sequence, Waypoint, Request, Macro, Step. There can be any number of elements at a lower level of the hierarchy, e.g., there can be any number of macros in a request. A waypoint is a location that a rover will traverse to. For a lander mission there is only one waypoint, the lander position. A request represents a high-level task. A macro, described in more detail below, is the functional element in WITS by which the user specifies commands and parameters. Macros have expansions into steps. A step is a low-level command that will be uplinked to the spacecraft. WITS can generate various format output sequences. There is a unique format for the FIDO rover. For a flight mission, WITS outputs sequences in the Spacecraft Activity Sequence File (SASF) format [8].

The Sequence window shows the sequences in one plan. Multiple sequences can be displayed. A plan generally represents the planning elements to generate one command sequence to be uplinked to the lander or rover. The sequences are shown on the right hand side of the Sequence window. Supporting multiple sequences is useful for integration of subsequences from different scientists or subsequences for different instruments into the final uplink sequence.

Waypoint sequence elements are automatically entered into the sequence when a waypoint is generated in one of the WITS views. When a waypoint sequence element is added, then a Request is also automatically added with it.

A list of macros which can be inserted into a sequence is shown on the left side of the Sequence window. Multiple lists of macros are available; choosing between macro lists is done via the pull-down menu above the macro list. A macro is inserted into a sequence by selecting the location in the sequence for it to be inserted and then double clicking on the macro in the macro list. Double clicking on a macro in the sequence causes the Macro window to pop up. The macro window for the FIDO IPS_scan macro is shown in Figure 6. The structure of the macro window is the same for all macros. At the top is a description area. The parameter names, values, and means for specifying their values is in the middle, and the macro expansion into steps is at the bottom. Depending on the type of parameter, text areas, pull-down menus, and slidebars are provided for specifying the parameter values. A macro-specific algorithm converts the parameters into the expansion which can have any number of steps.

A macro can generate View Objects which are displayed in the views to indicate what the macro is specifying. The IPS_scan macro generates a scan pattern for the FIDO mast-mounted point spectrometer. This macro generates View Objects to represent where the scan points are on the terrain. The 3x3 scan pattern is shown in both the Wedge and Panorama views of Figure 6.

There are various sequence editing features in the Sequence window, e.g., cut, copy, paste, and delete in the Action pull-down menu. Additionally, the user can click and drag an item in the sequence to another position in the sequence, e.g., the user can click on a macro and drag it into a different request. Waypoints can be reordered this way also.

5 Distributed Operations

The Sequence window File pull-down menu enables sequences to be loaded from the common server (e.g., located at JPL), or to be saved to the common server. This enables users to collaborate in sequence generation. Targets are also loaded from the server and saved to the server so that distributed users can see and utilize each other's targets. An example of distributed collaboration is to have scientists, who are distributed at different sites on the Internet, generate sequences which have only commands for the specific instruments for which they are the experts. They save their sequences to the common server and the core operations team can integrate their inputs into the final sequence by dragging requests out of their sequences into the final sequence to be sent to the rover. The distributed users can load the final sequence to see how their inputs were incorporated. Phone-conferencing and video-conferencing are useful additional means of communication during the collaboration process.

6 Implementation

The client WITS system which is used by the Internet-based users was implemented using the Java1.2 programming language. The system is accessed by remote users either by using the Java1.2 appletviewer or by going to a URL from a web browser. At this time, only browsers running on Windows95/98/NT operating systems can access WITS via a browser since Java1.2 is not yet supported for browsers on other operating systems. The two browsers which have been used are Netscape Navigator and Microsoft Internet Explorer. In this case, the user first must download the Java PlugIn in order for the browser to support Java1.2 language applets. Users who have computers with Sun Solaris operating systems download the Java Development Kit (JDK) onto their computers and access and run WITS from a URL using the JDK appletviewer application. It is anticipated that Netscape Navigator will soon support Java1.2 for the Solaris operating system, so those users will be able to access WITS via their browser as well.

The database is a structured Unix file system. Downlink data is organized by the location where the data was acquired and the instrument which took the data, e.g., results/sites/site-2/position-1/irPointSpec has data taken by the point spectrometer at position 1 of site 2. Data which relates to a whole site is placed at the site-N level, e.g., specifications for Overhead and Panorama views. The database has a plan directory which has all the files for a plan, e.g., the sequences. There is also a constants area which holds information which is constant over multiple plans, e.g., macro definitions.

7 Field Test Discussion

WITS was used as the ground operations system to generate command sequences for the FIDO rover in the April 1999 FIDO rover field test at Silver Lake, California. WITS was used both by operators in a trailer in the field local to the rover and by users distributed over the Internet. The field test operations architecture is shown in Figure 7. A server and client system were used by the operators in a trailer in the field. Satellite communication provided Internet access back to JPL. A server system at JPL provided

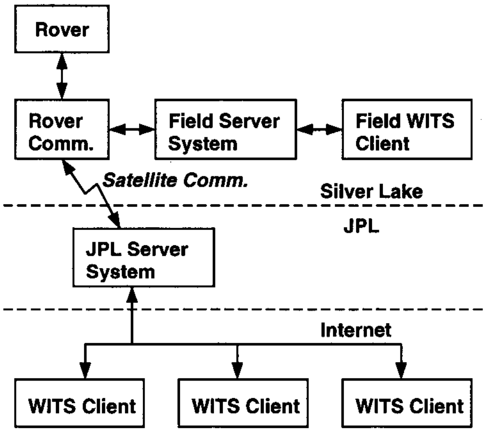

Figure 7: Field Test System Architecture

access by Internet based users.

The plan for the field test was to have sequences generated by operators in the trailer in the field and for users on the Internet to be able to view downlink data results via a server system at JPL. Two days at the end of the field test were set aside for operations by student teams over the Internet. A problem with the communication from the Rover Comm. system to the server system prevented Internet-based users from viewing downlink data over the Internet via WITS. The downlink from the Rover Comm. system to the two WITS server systems could not support downlink communication to two server systems. Therefore, only downlink of raw data to the field server system was done. The processed data in the field-located database was too large to download over the satellite link to JPL. Therefore, downlink data was not put on the JPL server system and Internet based users could not view downlink data via WITS during the field test until they were operators at the end of the field test.

High school students commanded the FIDO rover during the last two days of the field test using WITS via the Internet. High schools from Los Angeles, CA, Phoenix, AZ, Ithica, NY, and St. Louis, MO made up the LAPIS Student Tests Mission team. The Rover Comm. downlink system was switched to downlink to the JPL located server system to enable Internet-based users to view the downlink data and generated command sequences for the rover. The students viewed the downlink data and generated the command sequences. The field located operators ran WITS from the JPL located server as another Internet based user. After the students finished generating a sequence, the field located operators reviewed it and sent it for execution by the rover.

During field test operations from the trailer, WITS had the functionality to command eight different instruments (pancam, navcam, hazcams, bellycam, IPS, color microscopic imager, moessbauer, and mini-corer), as well as the mast, arm, and rover motions. Also, WITS had the functionality to display the downlink data results for all instruments in graphical and/or textural format. Some of the available functionality of WITS was not utilized in the field test due to field test time constraints. The pancam is a stereo camera pair on the rover's mast. Filters at 650nm, 750nm, and 850nm enable taking images at the three wavelengths. The images were combined in the downlink system to produce color images which were viewable in WITS. An initial problem in viewing the pancam images was that WITS encountered an out-of-memory error since a 111 stereo pair pancam panorama was taken at the start of the field test. WITS did not initially have enough memory to display 111 color images in the Panorama view. This was remedied by increasing the memory allocation to WITS. Also, the downlink processing system did not have accurate camera models for the pancam cameras so range data for the pancam images was not available in WITS.

The navcam is a monochrome camera pair on the rover mast. Images from the navcam were used in WITS for generating command sequences. The hazcams were monochrome stereo camera pairs mounted to the rover body. Hazcam images were displayed using the Wedge view. The bellycam was a monochrome stereo camera pair mounted on the underside of of the rover to view the instrument area under the rover. Wedge views were used to view bellycam images.

The IPS was the integrated point spectrometer which was mounted on the rover mast. The color microscopic imager was a close-up camera mounted on the rover instrument arm. The moessbauer was a spectrometer which as not mounted on the rover, but which analyzed rover acquired samples. The mini-corer was the drill mounted under the rover which drilled out rock cores.

8 Future Plans

Various capabilities are being added to WITS both to complete the suite of capabilities needed for actual mission operations and to overcome limitations identified from the FIDO field test. The new features will be demonstrated in future FIDO rover field tests. Limitations of WITS that became apparent during the field test included the lack of visualization of

Figure 8: 3D View

the rover position at a site, e.g., current rover position and orientation within the Overhead view, the lack of sequence simulation, and lack of 3D visualization. Current work is underway to display the rover's past and current position and orientation in the Overhead view. A 3D view, shown in Figure 8, is being developed using Java3D. The figure shows only the FIDO rover. 3D terrain visualization will be added to the 3D view as well. A companion Control window allows for interactively moving each articulated degree of freedom in the kinematic model. Simulation capability is being added to enable simulation of a sequence. During the simulation, the rover motion will be shown in the Panorama and Overhead views.

Capabilities necessary for mission use which are being added include encrypted communication, resource analysis, and rules checking. Encrypted communication will provide secure communication between the server system and clients. Resource analysis will compute the time, energy, and data volume used by each step of the sequence and compare the cumulative amounts with the allocations. Rules checking will check that specified rules are satisfied by the sequence, e.g., when a command must have been preceded in the sequence by another specific command.

9 Conclusions

WITS enables Internet-based distributed mission operations for future Mars rover missions. WITS is being reimplemented to move it from a research product to a flight mission product usable in actual Mars rover flight missions, e.g., the Mars'03 rover mission. Its use in the FIDO rover desert field test demonstrated its numerous capabilities and indicated features that still needed to be added.

Acknowledgements

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] Paul G. Backes, Gregory K. Tharp, and Kam S. Tso. The web interface for telescience (WITS). In *Proceedings IEEE International Conference on Robotics and Automation*, pages 411–417, Albuquerque, New Mexico, April 1997.

[2] Paul G. Backes, Kam S. Tso, and Gregory K. Tharp. Mars Pathfinder mission Internet-based operations using WITS. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 284–291, Leuven, Belgium, May 1998.

[3] S. Hayati, R. Volpe, P. Backes, J. Balaram, R. Welch, R. Ivlev, G. Tharp, S. Peters, T. Ohm, and R. Petras. The Rocky7 rover: A Mars sciencecraft prototype. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 2458–2464, Albuquerque, New Mexico, April 1997.

[4] R. Arvidson, Paul Backes, E. Baumgartner, D. Blaney, L. Dorsky, A. Haldemann, R. Lindemann, P. Schenker, and S. Squyers. Fido: Field-test rover for 2003 and 2005 mars sample return missions. In *30th Lunar and Planetary Science Conference*, Houston, Texas, March 15–19 1999.

[5] Eric Paulos and John Canny. Delivering real reality to the World Wide Web via telerobotics. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 1694–1699, Minneapolis, Minnesota, April 22-28 1996.

[6] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley. Robot teleoperation via WWW. In *Proceedings IEEE International Conference on Robotics and Automation*, May 1995.

[7] B. Cooper. Driving on the surface of Mars using the rover control workstation. In *SpaceOps98 Conference*, Tokyo, Japan, June 1998.

[8] W. Lombard. Spacecraft activity sequence file. Technical Report JPL Software Interface Specification Module (SFOC-1-SEQ-ANY-SASF), Revision 06, Jet Propulsion Laboratory, 1997.